

# IPKISS: A Parametric Design and Simulation Framework for Silicon Photonics

Wim Bogaerts, Pieter Dumon, Emmanuel Lambert, Martin Fiers, Shibnath Pathak, Antonio Ribeiro  
Photonics Research Group, Ghent University - imec, Center of Nano and Biophotonics, Belgium

**Abstract—** We present IPKISS, an open-source design environment for complex (silicon) photonic circuitry. The tool set allows flexible mask design, direct electromagnetic simulation and circuit models, and can be easily interfaced with existing tools.

## I. INTRODUCTION

UNTIL the advent of high-contrast silicon photonics, photonic integrated circuits (PICs) contained anything from a single component to a few hundred components. This in stark contrast to ULSI electronic circuits where millions or billions of components are integrated on a chip. Photonic circuit design is therefore still a largely manual activity, and automation of this process is obstructed by some complexities of photonics design: capturing a full spectrum, fast and slow nonlinearities and tolerances typically requires intensive simulation in different tools.

On the other hand, electronic design automation (EDA) tools support large-scale circuit design, but do not natively support photonics: As the electromagnetic time scale in photonics is much shorter than in electronics, design is often performed in the frequency domain; but the concepts of optical power, phase and wavelength are not easily transferred to electrical simulators, which often rely on Kirchhoff's laws. Still, the first photonic toolkits based on EDA environments have been demonstrated, with a limited support of photonic concepts (e.g. a few wavelengths) [1]. Also, some photonic simulators already interface to EDA tools, such as RSoft's OptSim.

Another difficulty is that photonics offer much larger variation in building blocks (waveguides, rings, interferometers, modulators, detectors, ...), and therefore a significant fraction of the design work is still performed at the physical level: many designs still require validation with full electromagnetic simulation.

We present the IPKISS design framework [2]. Unlike most commercial design tools, IPKISS is entirely based around a scripting environment: Using the industry-standard Python language, it is possible to design and simulate individual photonic components and hook them up in complex circuits. The open nature of the framework and the Python language make it possible to hook up third-party tools to extend the capability of the framework [3].

## II. MANY VIEWS ON ONE COMPONENT

IPKISS is based on a parametric design philosophy: A component is described in terms of parameters which can be

set by the designer. From this, different representations ('views') of the component can be derived. The most commonly used view is the mask layout, which breaks down the component in geometric primitives on a set of photomasks. This is hierarchical, so components can contain reusable subcomponents. Other views include input/output ports, an internal connectivity (netlist), an S-matrix formalism, etc. This is shown in Fig. 1.

In addition, views can be based on one another: for example, a 3D geometry of the component can be extracted from the layout ('virtual fabrication'). It is also possible to chain views together in an optimization loop to improve the functionality of a component [4]. All the information in the views is calculated only when needed (delayed execution) and cached for reuse.

IPKISS is a modular framework: parametric components can be reused, extended (in Python using well-known object-oriented principles such as subclassing) and even ported between technologies: component libraries can accept external technology information with default design rules (waveguide widths, bend radius, etc.) and fabrication information (waveguide cross sections, schematic process flow).

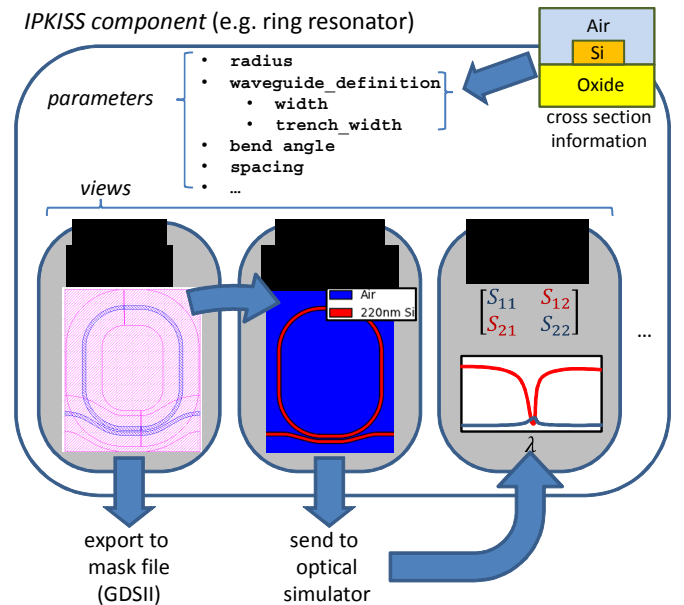


Fig. 1. Schematic internals of an IPKISS component. Based on the parameters, a layout can be generated. Other views, like a 2D/3D geometry or an S-matrix can be defined or automatically calculated from the layout.

This design philosophy is very similar to that of EDA, but because large-scale photonic circuit design is still far from mature the flexibility of IPKISS facilitates customization of designs.

While IPKISS is definitely not limited to photonics, it provides the most mature support for the design of silicon photonic circuits. Photonic ports, routing of waveguides along waypoints (taking into account bend radius and spacing) and modes in a waveguide are directly supported.

### III. INTEGRATED SIMULATION

One of the strengths of IPKISS is the ability to run optical simulations from within the design framework, without having to redefine the component in a separate tool. The virtual fabrication view can extract a 2D or 3D geometry from the mask layout, and this geometry can then be used for electromagnetic simulation.

For example, from the layout of the  $1 \times 2$  splitter in Fig. 2a we automatically generate the distribution of materials in Fig 2b. This is converted to a distribution of the dielectric constant  $\epsilon$  (Fig 2c) needed for MEEP, MIT's open-source FDTD solver [5]. Also, IPKISS knows the input and output ports of the device, so it can calculate the waveguide modes and use these as the excitation of the FDTD simulation (Fig 2d). From such a simulation, elements of an S-matrix of the component can be extracted, and stored for later use. Support for this simulation flow is built into the IPKISS framework.

But FDTD is not necessarily the best simulation method for

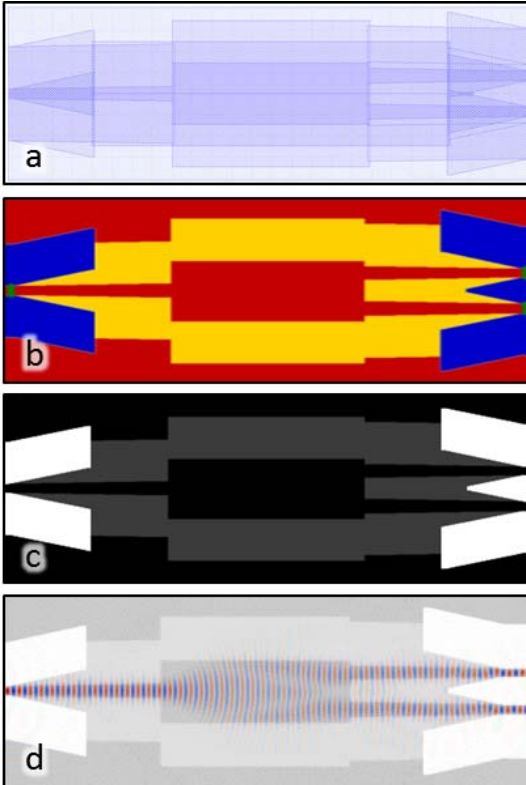


Fig 2: Different views on the same component. (a) mask layout, (b) automatically generated 3D geometry, (c) dielectric constant distribution for FDTD-simulation in MEEP, (d) Snapshot of electric field profile in MEEP.

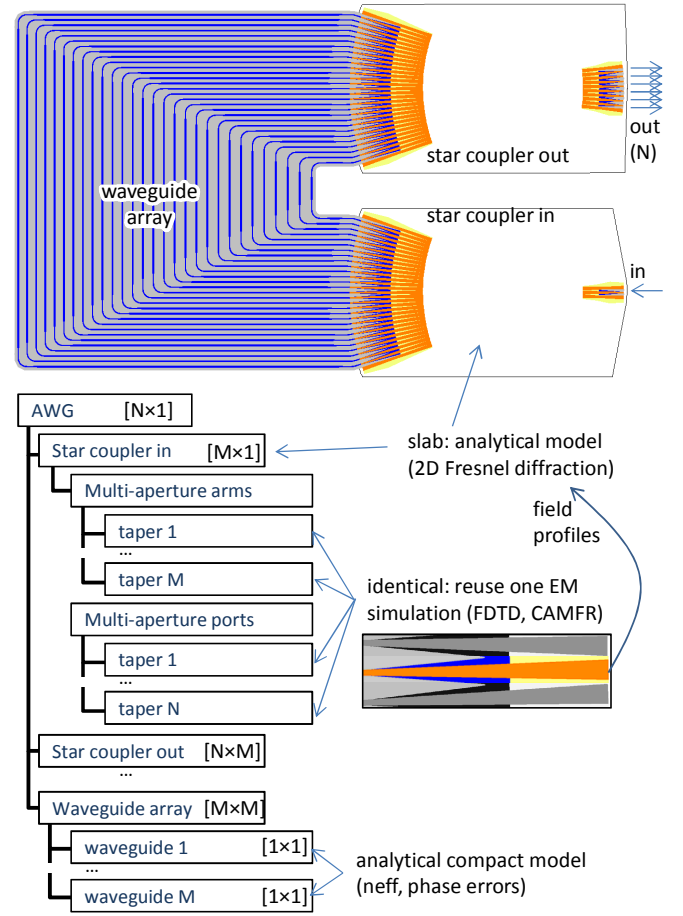


Fig. 3: Example of an AWG model implemented in IPKISS. The hierarchy of the AWG is exploited in the model, as each block is simulated with its own strategy.

all types of components. IPKISS supports hierarchical modeling: every component can have different modeling strategies, and these can be hooked together in a circuit model.

Figure 3 shows an example of models we have combined to simulate an arrayed waveguide grating (AWG) [6]. The array waveguides are simulated with a compact behavioral model incorporating loss, phase delay and phase error statistics. The free propagation region is a uniform slab waveguide, and is simulated using analytical 2D Fresnel diffraction. The tapered regions at the waveguide-slab interface are modeled numerically, either in MEEP, or with eigenmode propagation in CAMFR [7]. The parts are hooked up using hierarchical T-matrix multiplication (assuming linearity). IPKISS's delayed execution will trigger these simulations only when needed.

The physical simulations can be wrapped in compact models for large circuit simulations. This can scale to thousands of components, depending on the choice of circuit simulator: one example is the use of a Verilog A representation for Spice-based simulation, but this limits the photonic concepts that can be incorporated in the model.

Alternatively, we have interfaced IPKISS with the Caphe circuit simulator [9]. This tool, developed in our group, is based on scatter matrices, and can efficiently simulate large photonic circuits in both frequency domain and time domain.

#### IV. TOOL INTEGRATION

IPKISS itself contains very few simulation algorithms. Instead, it is an open framework designed to harness the strengths of tools that are already available. The use of Python helps a lot: easy to learn, very readable, and it comes with a many scientific libraries. It also provides various mechanisms to integrate third-party libraries. For example, we integrated with MEEP by defining a Python interface to MEEP [8].

Interfaces can also go the other way: third-party tools can access components in IPKISS. This is especially useful for EDA environments: in the view of stronger photonic-electronic integration, an efficient interface between EDA and photonic tools will be a key enabler for silicon photonics.

Therefore, we implemented a proof-of-concept interface between the Cadence design framework and IPKISS. Using the industry-standard OpenAccess framework [10], we managed to instantiate IPKISS building blocks from Cadence, and open the mask layout in the Virtuoso editor. Also, we associated a compact model written in VerilogA with our component and simulate that in Spectre, a Spice-based simulator.

While IPKISS could be used to launch physical simulations from within an EDA tool, this is not desirable when running a circuit simulation, this would be too time consuming. But it allows the designer to optimize building blocks from the same environment as the large-scale circuit design.

IPKISS is not limited to just the design process. We extended the framework to control our measurement hardware, and we can incorporate test and measurement information in our design. For instance, we can extract the positions of the optical input and output ports (using vertical grating couplers [11]) from the mask layout to program a probe station equipped with high-precision fiber position stages.

#### V. ACCESS TO IPKISS

In April 2012, IPKISS was released as open source, under the GNU General Public License (v2) [12], and the code base can be downloaded freely [13]. This open model is intended to stimulate developments on the framework and provide the photonics community with an accessible tool set for complex design.

IPKISS was originally conceived as a tool for generating complex photonic mask layouts, but it evolved into the flexible simulation framework presented here. Over the last 10 years, it has been actively used in tens of photomasks. IPKISS is now in use by over 50 research groups and companies worldwide, and is also supported by the silicon multi-project-wafer service ePIXfab [15],[14] in the form of Process Design Kits (PDK).

#### VI. SUMMARY

We presented the IPKISS design framework and illustrate how it can facilitate the design process for silicon photonic circuits. The open-source framework allows mask design, electromagnetic and circuit simulation from a single

component definition, and the framework can be extended to interface with third-party tools, for example MIT's MEEP or OpenAccess compliant EDA tools.

#### ACKNOWLEDGEMENTS

The proof-of-concept integration with OpenAccess was supported by the European Union through the FP7-ICT-HELIOS project.

#### REFERENCES

- [1] A. Mekis et al. "Monolithic Integration Photonic and Electronic Circuits in a CMOS Process", Proc. of SPIE Vol. 6897, 68970L, (2008)
- [2] The IPKISS web site: [www.ipkiss.org](http://www.ipkiss.org)
- [3] P. Bienstman et al. "Python in Nanophotonics Research", Comp. Sci. Eng 9(3), p.46 (2007)
- [4] W. Bogaerts et al., "Closed-loop modeling of silicon nanophotonics from design to fabrication and back again", Opt. Quantum Electron., 40(11), p.801 (2009)
- [5] A.F. Oskooi et al., "MEEP: A Flexible Free-Software Package for Electromagnetic Simulations by the FDTD Method," *Computer Physics Comm.* 181(3), pp. 687. (2010)
- [6] S. Pathak et al., "Compact SOI-Based AWG With Flattened Spectral Response Using a MMI", 8th International Conference in Group IV Photonics 2011, p.45 (WC2) (2011)
- [7] P. Bienstman et al. "Optical modelling of photonic crystals and VCSELs using eigenmode expansion and perfectly matched layers", Opt. Quantum Electron. 33, p.327 (2001)
- [8] E. Lambert et al. "Python bindings for the open source electromagnetic simulator MEEP", *Comp. Sci. Eng.* 13(3), p.53 (2011)
- [9] M. Fiers et al. "Time-domain and frequency-domain modeling of nonlinear optical components on circuit-level using a node-based approach", *J. Opt. Soc. Am. B*, 29(5), p.896 (2011)
- [10] T. Blanchard et al. "The OpenAccess coalition - the drive to an open industry standard information model, API, and reference implementation for IC design data", Qual. Electron. Des., p. 69 (2002)
- [11] D. Vermeulen et al. "High-efficiency fiber-to-chip grating couplers realized using an advanced CMOS-compatible silicon-on-insulator platform", Opt. Express, 18(17), p.18278 (2010)
- [12] GNU general public license v2: [www.gnu.org/licenses/gpl-2.0.html](http://www.gnu.org/licenses/gpl-2.0.html)
- [13] IPKISS code repository on Codebase HQ [ipkiss.codebasehq.com/projects/ipkiss](http://ipkiss.codebasehq.com/projects/ipkiss)
- [14] P. Dumon et al. "Towards foundry approach for silicon photonics: silicon photonics platform ePIXfab", Electron. Lett., 45(12), p.581 (2009)
- [15] ePIXfab web site: [www.epixfab.eu](http://www.epixfab.eu)